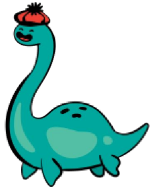
Iceberg Views

An Introduction

Eduard Tudenhoefner - OSS@Dremio

April 2022

Agenda



- Quick Intro to Apache Iceberg
- What are Views?
- Iceberg Views
- Ongoing & Future work

Apache Iceberg

Apache Iceberg - https://iceberg.apache.org

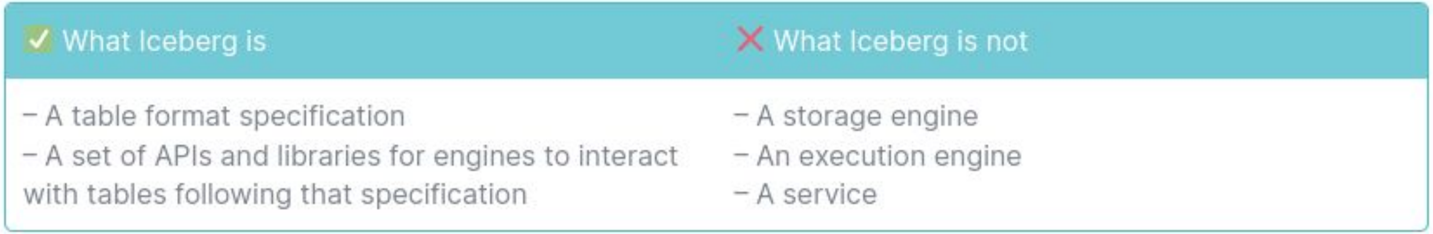
- Open table format that supports huge analytic datasets
- engine-agnostic format that is understood by different **compute engines**
    - Spark / Dremio / Trino / Presto / Flink / …

Apache Iceberg - https://iceberg.apache.org

- Open table format that supports huge analytic datasets
- engine-agnostic format that is understood by different **compute engines**
  - Spark / Dremio / Trino / Presto / Flink / …
- main features are:
  - Time Travel
  - Schema evolution (without side-effects)
  - Partition evolution
  - Transactions
  - great performance (built for huge datasets)
  - works with different cloud storage providers

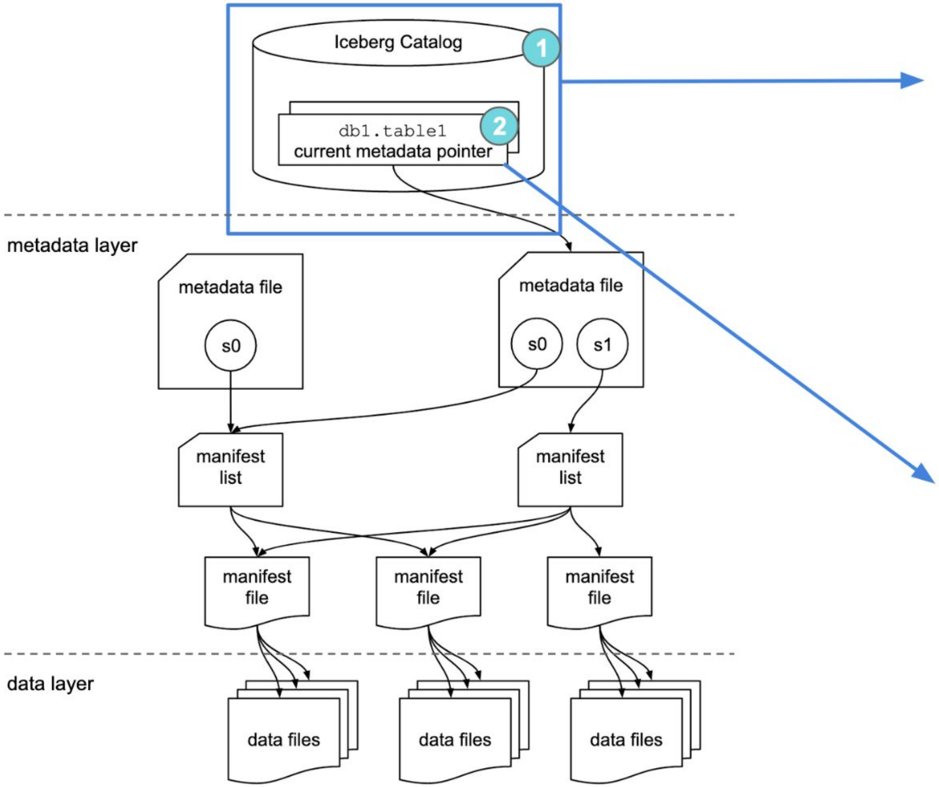
Apache Iceberg - https://iceberg.apache.org

- Open table format that supports huge analytic datasets
- engine-agnostic format that is understood by different **compute engines**
  - Spark / Dremio / Trino / Presto / Flink / …
- main features are:
  - Time Travel
  - Schema evolution (without side-effects)
  - Partition evolution
  - Transactions
  - great performance (built for huge datasets)
  - works with different cloud storage providers



✅ What Iceberg is	❌ What Iceberg is not
– A table format specification – A set of APIs and libraries for engines to interact with tables following that specification	– A storage engine – An execution engine – A service

Apache Iceberg - https://iceberg.apache.org



Iceberg Catalog

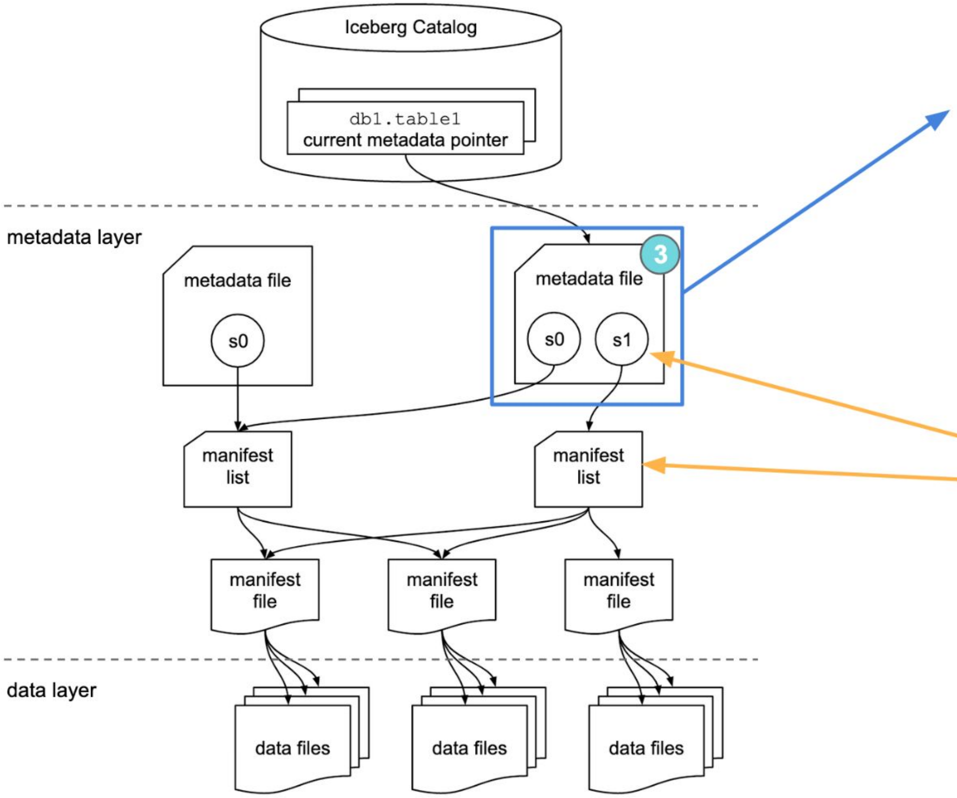
- A store that houses the current metadata pointer for Iceberg tables
- Must support atomic operations for updating the current metadata pointer (e.g. HDFS, HMS, Nessie)

`table1`'s current metadata pointer

- Mapping of table name to the location of current metadata file

https://www.dremio.com/resources/guides/apache-iceberg-an-architectural-look-under-the-covers/

Apache Iceberg - https://iceberg.apache.org

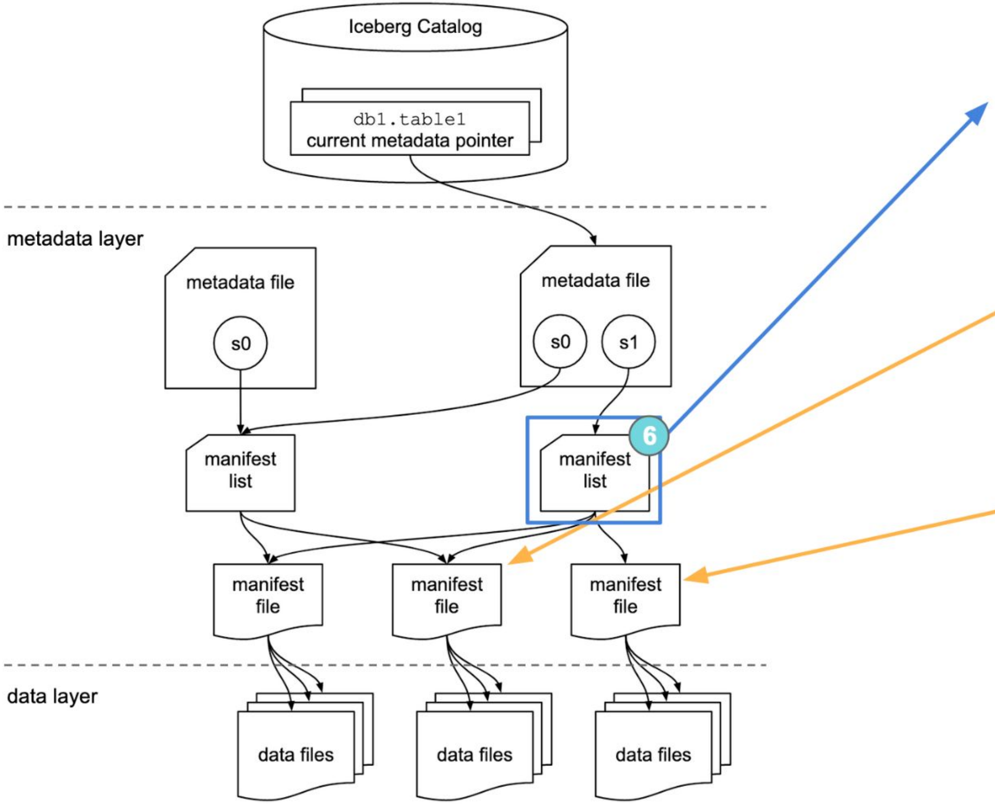


Metadata file - stores metadata about a table at a certain point in time

```
{
  "table-uuid" : "<uuid>",
  "location" : "/path/to/table/dir",
  "schema": {...},
  "partition-spec": [ {<partition-details>}, ...],
4 "current-snapshot-id": <snapshot-id>,
  "snapshots": [ {
      "snapshot-id": <snapshot-id>
5     "manifest-list": "/path/to/manifest/list.avro"
  }, ...],
  ...
}
```

https://www.dremio.com/resources/guides/apache-iceberg-an-architectural-look-under-the-covers/

Apache Iceberg - https://iceberg.apache.org

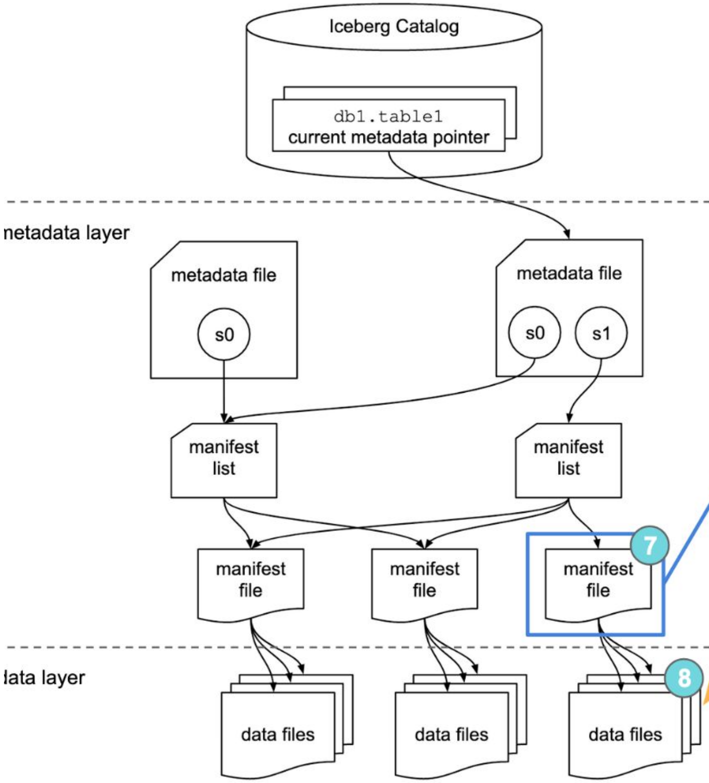


Manifest list file - a list of manifest files

```
{
  "manifest-path" : "/path/to/manifest/file.avro",
  "added-snapshot-id": <snapshot-id>,
  "partition-spec-id": <partition-spec-id>,
  "partitions": [ {partition-info}, ...],
  ...
}
{
  "manifest-path" : "/path/to/manifest/file2.avro",
  "added-snapshot-id": <snapshot-id>,
  "partition-spec-id": <partition-spec-id>,
  "partitions": [ {partition-info}, ...],
  ...
}
```

https://www.dremio.com/resources/guides/apache-iceberg-an-architectural-look-under-the-covers/

Apache Iceberg - https://iceberg.apache.org



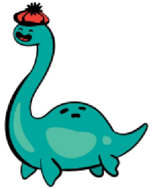
Manifest file - a list of data files, along with details and stats about each data file

```
{
  "data-file": {
    "file-path": "/path/to/data/file.parquet",
    "file-format": "PARQUET",
    "partition": {"<part-field>":{"<data-type>":<value>}},
    "record-count": <num-records>,
    "null-value-counts": [{
      "column-index": "1", "value": 4
    }, ...],
    "lower-bounds": [{
      "column-index": "1", "value": "aaa"
    }, ...],
    "upper-bounds": [{
      "column-index": "1", "value": "eee"
    }, ...],
  }
  ...
}
{
 ...
}
```

https://www.dremio.com/resources/guides/apache-iceberg-an-architectural-look-under-the-covers/

What are Views?

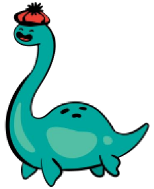
What are Views?

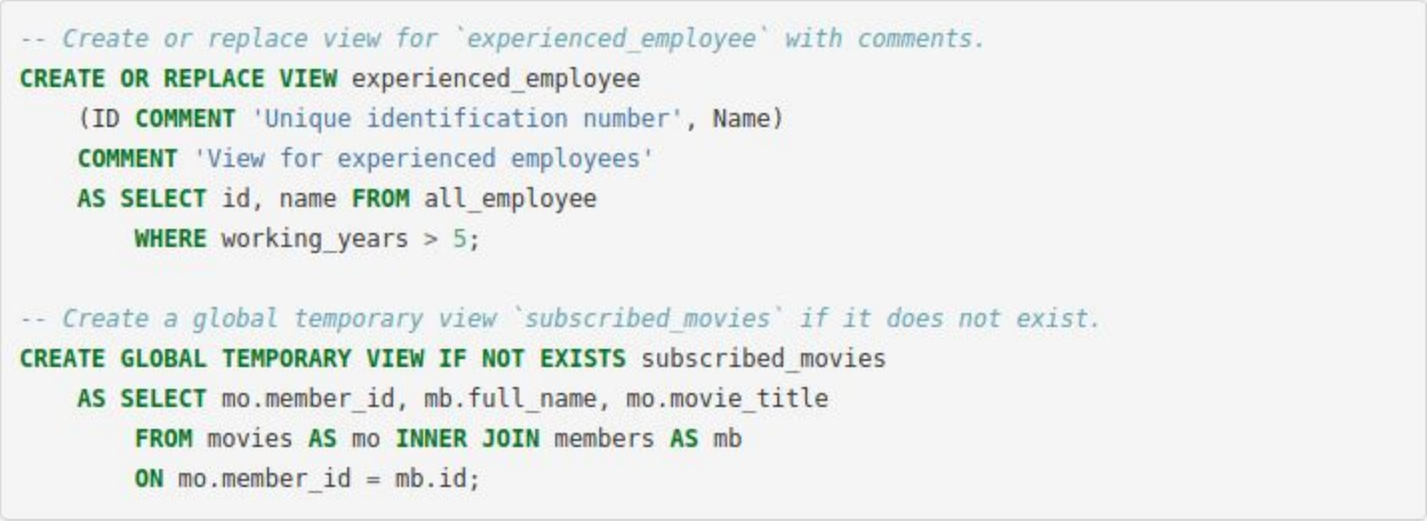


- virtual table that contains real rows/cols of an actual table
- query definition is executed whenever view is accessed
- think: **logical view of data**

+ will always return latest data
- performance depends on how good the SQL is the view depends on

Spark View Example



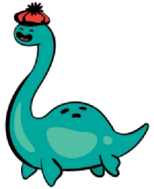


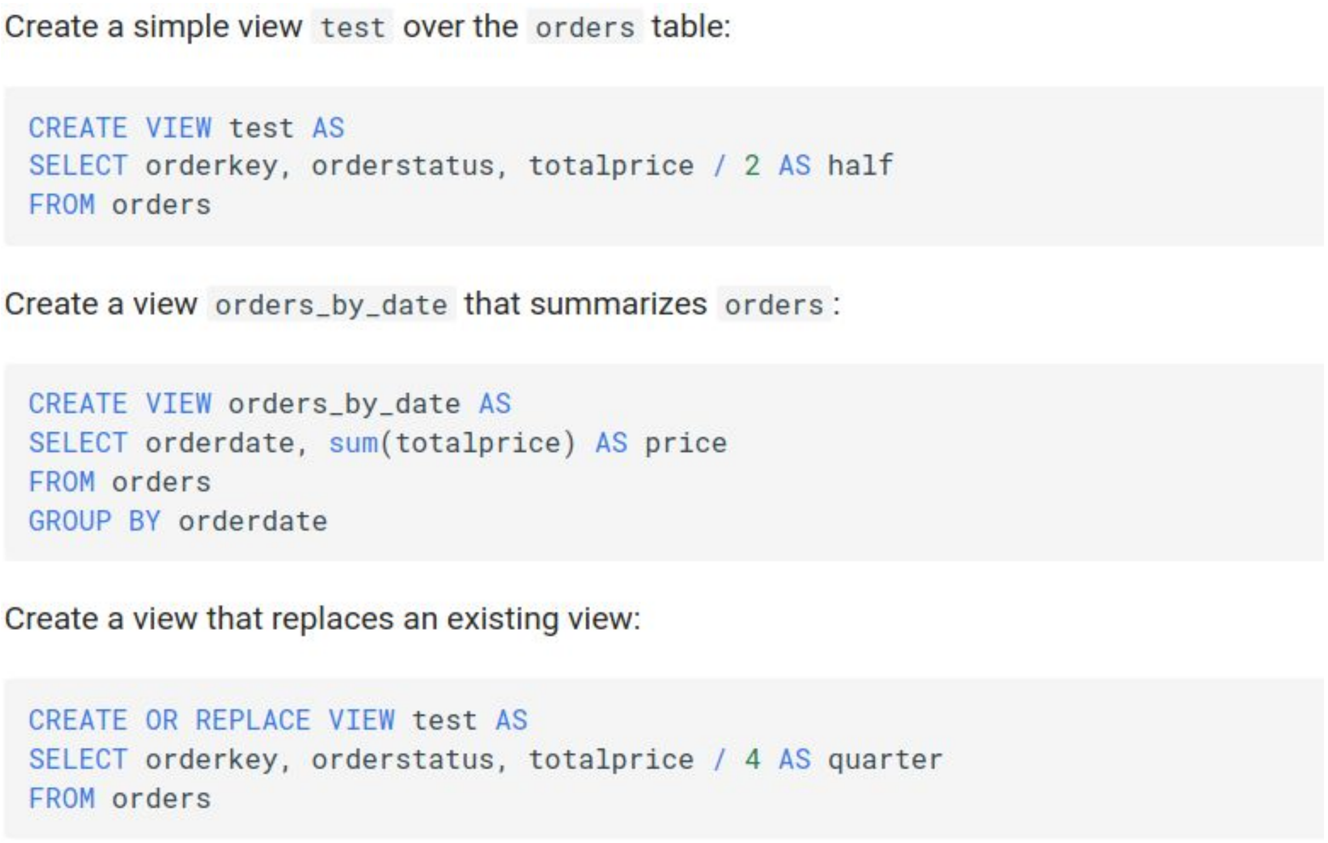
```
-- Create or replace view for `experienced_employee` with comments.
CREATE OR REPLACE VIEW experienced_employee
    (ID COMMENT 'Unique identification number', Name)
    COMMENT 'View for experienced employees'
    AS SELECT id, name FROM all_employee
        WHERE working_years > 5;

-- Create a global temporary view `subscribed_movies` if it does not exist.
CREATE GLOBAL TEMPORARY VIEW IF NOT EXISTS subscribed_movies
    AS SELECT mo.member_id, mb.full_name, mo.movie_title
        FROM movies AS mo INNER JOIN members AS mb
        ON mo.member_id = mb.id;
```

https://spark.apache.org/docs/3.2.1/sql-ref-syntax-ddl-create-view.html

Trino View Example





Create a simple view `test` over the `orders` table:

```
CREATE VIEW test AS
SELECT orderkey, orderstatus, totalprice / 2 AS half
FROM orders
```

Create a view `orders_by_date` that summarizes `orders`:

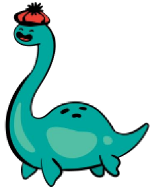
```
CREATE VIEW orders_by_date AS
SELECT orderdate, sum(totalprice) AS price
FROM orders
GROUP BY orderdate
```

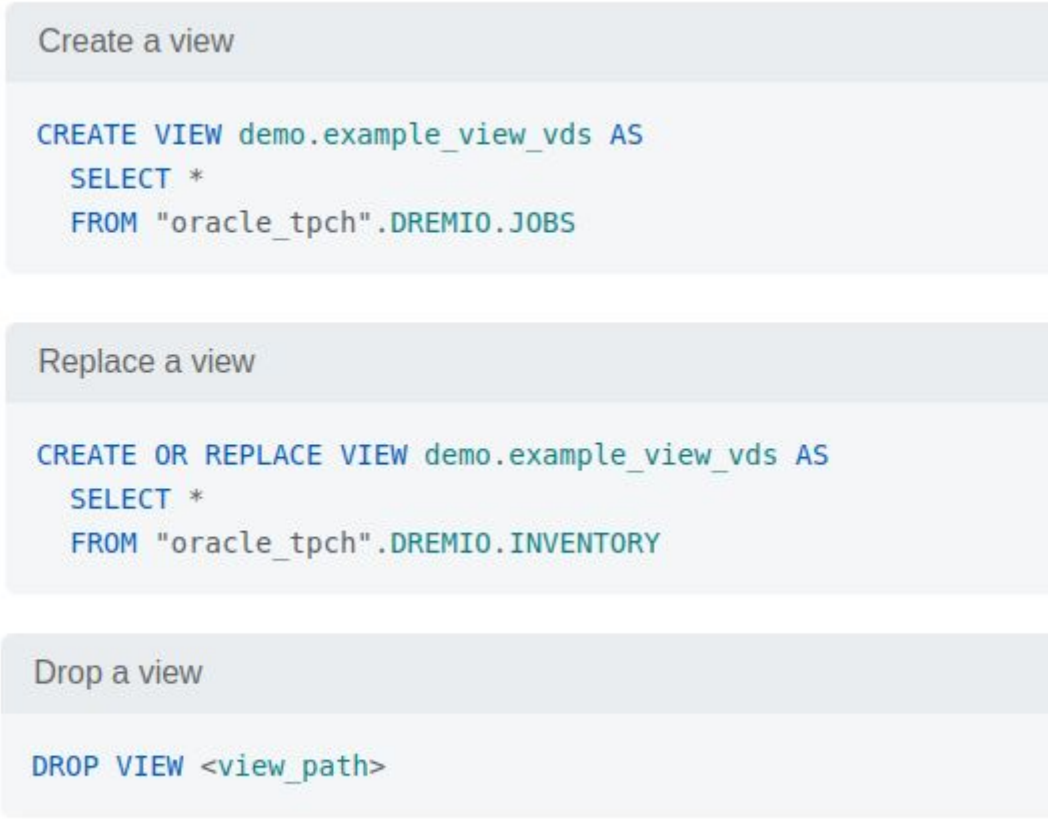
Create a view that replaces an existing view:

```
CREATE OR REPLACE VIEW test AS
SELECT orderkey, orderstatus, totalprice / 4 AS quarter
FROM orders
```

https://trino.io/docs/current/sql/create-view.html

Dremio View Example





Create a view

```
CREATE VIEW demo.example_view_vds AS
  SELECT *
  FROM "oracle_tpch".DREMIO.JOBS
```

Replace a view

```
CREATE OR REPLACE VIEW demo.example_view_vds AS
  SELECT *
  FROM "oracle_tpch".DREMIO.INVENTORY
```

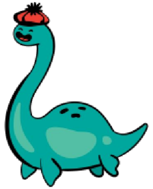
Drop a view

```
DROP VIEW <view_path>
```

https://docs.dremio.com/cloud/sql/commands/views/

Iceberg Views

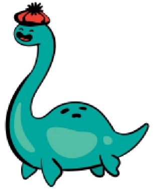
The Road to Iceberg Views



- Spark / Trino / Dremio / ... support Views, but View metadata is specific to engine
- Motivation:
  - Views created in one engine cannot be read/altered by another engine
  - even if engines share the same metastore

- overall goals:
  - Views that are **cross-compatible** across engines
  - Versioning of Views

The Road to Iceberg Views

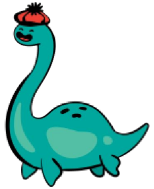


1. allow only ANSI-compliant SQL and anything that is truly common across all engines

2. add a "dialect" field to the View metadata

3. store AST produced by Calcite / create IR to represent Views

The Road to Iceberg Views

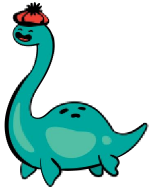


1 allow only ANSI-compliant SQL and anything that is truly common across all engines

2 add a "dialect" field to the View metadata

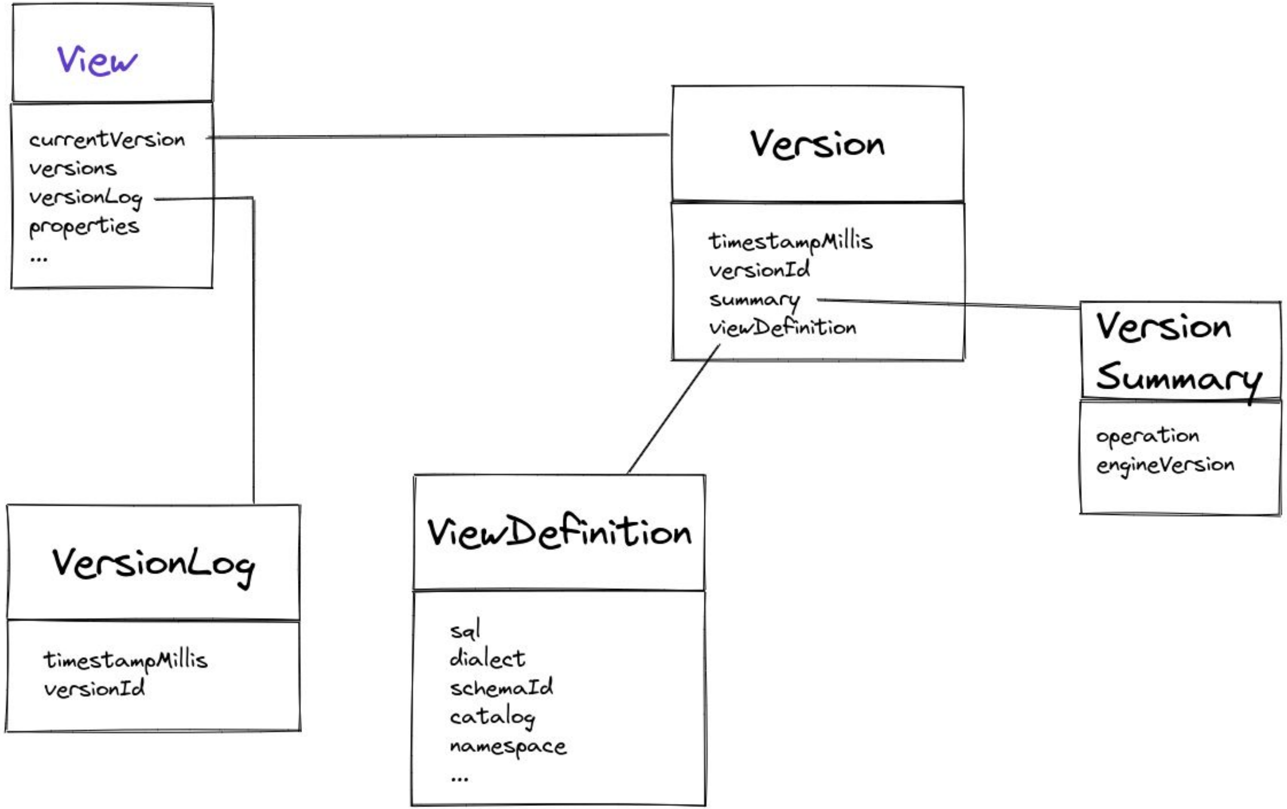
3 store AST produced by Calcite / create IR to represent Views

Iceberg View Metadata & Spec

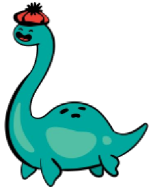


- added by Netflix team in https://github.com/apache/iceberg/pull/3188
- https://github.com/apache/iceberg/blob/master/format/view-spec.md
- modeled after Iceberg Tables
- View metadata is maintained in metadata files
- every change creates a new view metadata file
- each metadata file is self-sufficient and tracks history of changes to DDL of view

View Architecture

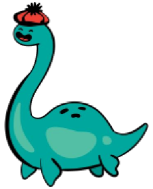


Example



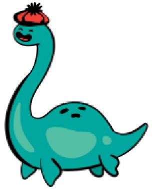
- Imagine the following sequence of operations:
  - `1. CREATE TABLE base_tab(c1 int, c2 varchar);`
  - `2. INSERT INTO base_tab VALUES (1,’one’), (2,’two’);`
  - `3. CREATE VIEW common_view AS SELECT * FROM base_tab;`
  - `4. CREATE OR REPLACE VIEW common_view AS SELECT count(*) AS my_cnt FROM base_tab;`

```
3. CREATE VIEW common_view AS SELECT * FROM base_tab;
```



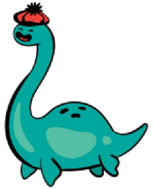
3. CREATE VIEW common_view AS SELECT * FROM base_tab;

```
{
  "format-version" : 1,        => JSON format. Will change as format evolves.
  "location" : "s3n://my_company/my/warehouse/anorwood.db/common_view",
  "current-version-id" : 1, => current / latest version of the view. '1' here since this metadata was created when the view was created.
  "properties" : {           => shows properties of the view
    "comment" : "View captures all the data from the table"  => View comment
  },
```



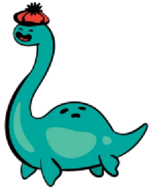
3. CREATE VIEW common_view AS SELECT * FROM base_tab;

```
{
  "format-version" : 1,        => JSON format. Will change as format evolves.
  "location" : "s3n://my_company/my/warehouse/anorwood.db/common_view",
  "current-version-id" : 1, => current / latest version of the view. '1' here since this metadata was created when the view was created.
  "properties" : {              => shows properties of the view
    "comment" : "View captures all the data from the table"  => View comment
  },
  "versions" : [ {              => Last few versions of the view.
    "version-id" : 1,
    "parent-version-id" : -1,
    "timestamp-ms" : 1573518431292,
    "summary" : {
      "operation" : "create",          => View operation that caused this metadata to be created
      "engineVersion" : "presto-350", => Version of the engine that performed the operation (create / replace)
    },
```

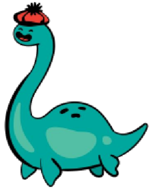


3. CREATE VIEW common_view AS SELECT * FROM base_tab;

```
{
  "format-version" : 1,         => JSON format. Will change as format evolves.
  "location" : "s3n://my_company/my/warehouse/anorwood.db/common_view",
  "current-version-id" : 1, => current / latest version of the view. '1' here since this metadata was created when the view was created.
  "properties" : {              => shows properties of the view
    "comment" : "View captures all the data from the table"  => View comment
  },
  "versions" : [ {              => Last few versions of the view.
    "version-id" : 1,
    "parent-version-id" : -1,
    "timestamp-ms" : 1573518431292,
    "summary" : {
      "operation" : "create",          => View operation that caused this metadata to be created
      "engineVersion" : "presto-350", => Version of the engine that performed the operation (create / replace)
    },
    "representations" : [ {     => SQL metadata of the view
      "type" : "sql",
      "sql" : "SELECT *\nFROM\n  base_tab\n",  => original view SQL
      "dialect" : "spark",
      "schema-id" : 1,
      "default-catalog" : "iceberg",
      "default-namespace" : [ "anorwood" ]
    } ],
  } ],
```

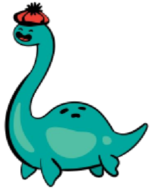


3. CREATE VIEW common_view AS SELECT * FROM base_tab;



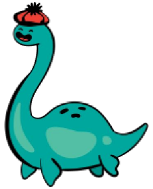
```
{
  "format-version" : 1,      => JSON format. Will change as format evolves.
  "location" : "s3n://my_company/my/warehouse/anorwood.db/common_view",
  "current-version-id" : 1, => current / latest version of the view. '1' here since this metadata was created when the view was created.
  "properties" : {           => shows properties of the view
    "comment" : "View captures all the data from the table"  => View comment
  },
  "versions" : [ {            => Last few versions of the view.
    "version-id" : 1,
    "parent-version-id" : -1,
    "timestamp-ms" : 1573518431292,
    "summary" : {
      "operation" : "create",          => View operation that caused this metadata to be created
      "engineVersion" : "presto-350", => Version of the engine that performed the operation (create / replace)
    },
    "representations" : [ {     => SQL metadata of the view
      "type" : "sql",
      "sql" : "SELECT *\nFROM\n  base_tab\n", => original view SQL
      "dialect" : "spark",
      "schema-id" : 1,
      "default-catalog" : "iceberg",
      "default-namespace" : [ "anorwood" ]
    } ],
  } ],
  "version-log" : [ {   => Log of the created versions
    "timestamp-ms" : 1573518431292,
    "version-id" : 1
  } ],
```

3. CREATE VIEW common_view AS SELECT * FROM base_tab;



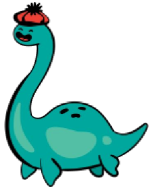
```
{
  "format-version" : 1,      => JSON format. Will change as format evolves.
  "location" : "s3n://my_company/my/warehouse/anorwood.db/common_view",
  "current-version-id" : 1, => current / latest version of the view. '1' here since this metadata was created when the view was created.
  "properties" : {           => shows properties of the view
    "comment" : "View captures all the data from the table"   => View comment
  },
  "versions" : [ {           => Last few versions of the view.
    "version-id" : 1,
    "parent-version-id" : -1,
    "timestamp-ms" : 1573518431292,
    "summary" : {
      "operation" : "create",          => View operation that caused this metadata to be created
      "engineVersion" : "presto-350", => Version of the engine that performed the operation (create / replace)
    },
    "representations" : [ {     => SQL metadata of the view
      "type" : "sql",
      "sql" : "SELECT *\nFROM\n  base_tab\n",  => original view SQL
      "dialect" : "spark",
      "schema-id" : 1,
      "default-catalog" : "iceberg",
      "default-namespace" : [ "anorwood" ]
    } ],
  } ],
  "version-log" : [ {    => Log of the created versions
    "timestamp-ms" : 1573518431292,
    "version-id" : 1
  } ],
  "schemas": [ {         => Schema of the view expressed in Iceberg types
    "schema-id": 1,
    "type" : "struct",
    "fields" : [ {
      "id" : 0,
      "name" : "c1",
      "required" : false,
      "type" : "int",
      "doc" : ""
    }, {
      "id" : 1,
      "name" : "c2",
      "required" : false,
      "type" : "string",
      "doc" : ""
    } ]
  } ],
  "current-schema-id": 1
}
```

```
4. CREATE OR REPLACE VIEW common_view AS SELECT count(*) AS my_cnt FROM base_tab;
```



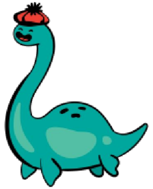
4. CREATE OR REPLACE VIEW common_view AS SELECT count(*) AS my_cnt FROM base_tab;

```
{
  "format-version" : 1,
  "location" : "s3n://my_company/my/warehouse/anorwood.db/common_view",
  "current-version-id" : 2,     => Version 2 was created
  "properties" : {
    "comment" : "View captures count of the data from the table"
  },
```

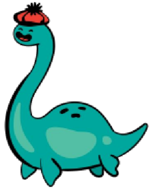


4. CREATE OR REPLACE VIEW common_view AS SELECT count(*) AS my_cnt FROM base_tab;

```
{
  "format-version" : 1,
  "location" : "s3n://my_company/my/warehouse/anorwood.db/common_view",
  "current-version-id" : 2,     => Version 2 was created
  "properties" : {
    "comment" : "View captures count of the data from the table"
  },
  "versions" : [ {
    "version-id" : 1,
    "parent-version-id" : -1,
    "timestamp-ms" : 1573518431292,
    "summary" : {
      "operation" : "create",
      "engineVersion" : "presto-350",
    },
    "representations" : [ {
      "type" : "sql",
      "sql" : "SELECT *\nFROM\n  base_tab\n",
      "dialect" : "spark",
      "schema-id" : 1,
      "default-catalog" : "iceberg",
      "default-namespace" : [ "anorwood" ]
    } ],
    "properties" : { }
  }, {
```

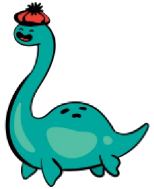


4. CREATE OR REPLACE VIEW common_view AS SELECT count(*) AS my_cnt FROM base_tab;

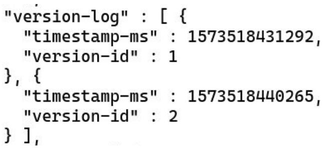


```
{
  "format-version" : 1,
  "location" : "s3n://my_company/my/warehouse/anorwood.db/common_view",
  "current-version-id" : 2,     => Version 2 was created
  "properties" : {
    "comment" : "View captures count of the data from the table"
  },
  "versions" : [ {
    "version-id" : 1,
    "parent-version-id" : -1,
    "timestamp-ms" : 1573518431292,
    "summary" : {
      "operation" : "create",
      "engineVersion" : "presto-350",
    },
    "representations" : [ {
      "type" : "sql",
      "sql" : "SELECT *\nFROM\n  base_tab\n",
      "dialect" : "spark",
      "schema-id" : 1,
      "default-catalog" : "iceberg",
      "default-namespace" : [ "anorwood" ]
    } ],
    "properties" : { }
  }, {
    "version-id" : 2,                      => new version
    "parent-version-id" : 1,               => Version 2 was created on top of version 1, making parent-version-id 1
    "timestamp-ms" : 1573518440265,
    "summary" : {
      "operation" : "replace",             => The 'replace' operation caused this latest version creation
      "engineVersion" : "spark-2.4.4",     => Spark engine created this update
    },
```

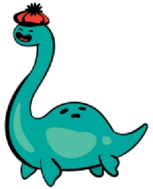
4. CREATE OR REPLACE VIEW common_view AS SELECT count(*) AS my_cnt FROM base_tab;

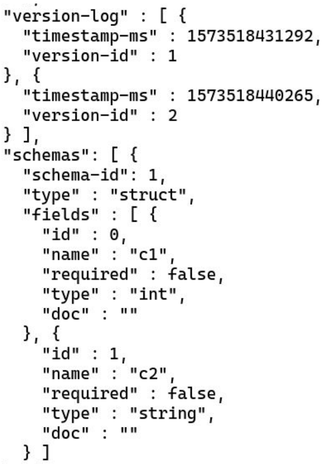


```
{
  "format-version" : 1,
  "location" : "s3n://my_company/my/warehouse/anorwood.db/common_view",
  "current-version-id" : 2,     => Version 2 was created
  "properties" : {
    "comment" : "View captures count of the data from the table"
  },
  "versions" : [ {
    "version-id" : 1,
    "parent-version-id" : -1,
    "timestamp-ms" : 1573518431292,
    "summary" : {
      "operation" : "create",
      "engineVersion" : "presto-350",
    },
    "representations" : [ {
      "type" : "sql",
      "sql" : "SELECT *\nFROM\n  base_tab\n",
      "dialect" : "spark",
      "schema-id" : 1,
      "default-catalog" : "iceberg",
      "default-namespace" : [ "anorwood" ]
    } ],
    "properties" : { }
  }, {
    "version-id" : 2,                    => new version
    "parent-version-id" : 1,             => Version 2 was created on top of version 1, making parent-version-id 1
    "timestamp-ms" : 1573518440265,
    "summary" : {
      "operation" : "replace",           => The 'replace' operation caused this latest version creation
      "engineVersion" : "spark-2.4.4",   => Spark engine created this update
    },
    "representations" : [ {
      "type" : "sql",
      "sql" : "SELECT \"count\"(*) my_cnt\nFROM\n  base_tab\n",  => Note the updated text from the 'replace' view statement
      "dialect" : "spark",
      "schema-id" : 2,
      "default-catalog" : "iceberg",
      "default-namespace" : [ "anorwood" ]
    },
```

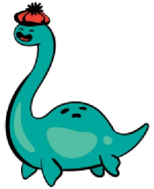


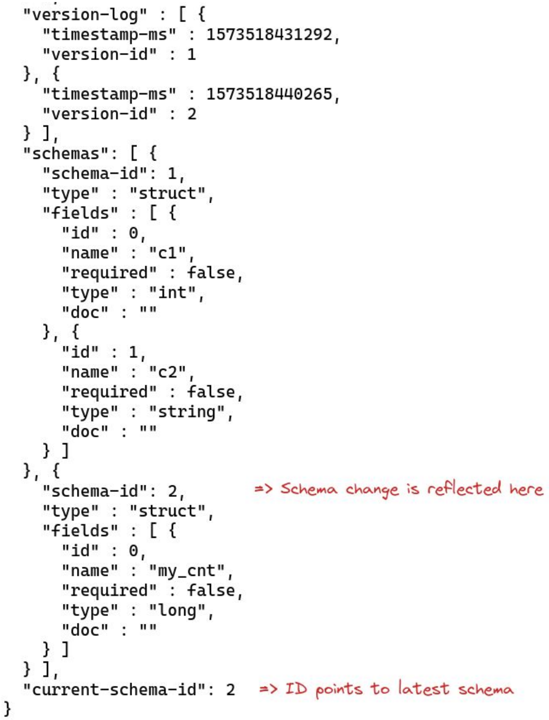
```
"version-log" : [ {
  "timestamp-ms" : 1573518431292,
  "version-id" : 1
}, {
  "timestamp-ms" : 1573518440265,
  "version-id" : 2
} ],
```



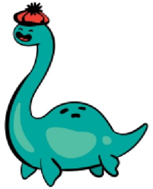


```
"version-log" : [ {
  "timestamp-ms" : 1573518431292,
  "version-id" : 1
}, {
  "timestamp-ms" : 1573518440265,
  "version-id" : 2
} ],
"schemas": [ {
  "schema-id": 1,
  "type" : "struct",
  "fields" : [ {
    "id" : 0,
    "name" : "c1",
    "required" : false,
    "type" : "int",
    "doc" : ""
  }, {
    "id" : 1,
    "name" : "c2",
    "required" : false,
    "type" : "string",
    "doc" : ""
  } ]
```





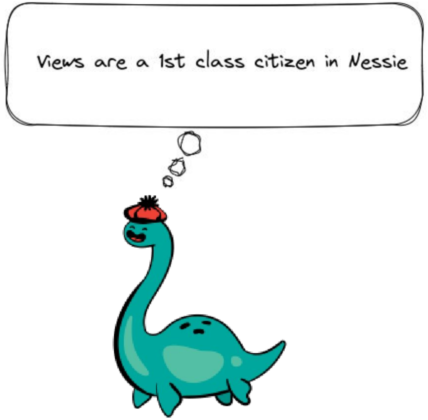
```
  "version-log" : [ {
    "timestamp-ms" : 1573518431292,
    "version-id" : 1
  }, {
    "timestamp-ms" : 1573518440265,
    "version-id" : 2
  } ],
  "schemas": [ {
    "schema-id": 1,
    "type" : "struct",
    "fields" : [ {
      "id" : 0,
      "name" : "c1",
      "required" : false,
      "type" : "int",
      "doc" : ""
    }, {
      "id" : 1,
      "name" : "c2",
      "required" : false,
      "type" : "string",
      "doc" : ""
    } ]
  }, {
    "schema-id": 2,        => Schema change is reflected here
    "type" : "struct",
    "fields" : [ {
      "id" : 0,
      "name" : "my_cnt",
      "required" : false,
      "type" : "long",
      "doc" : ""
    } ]
  } ],
  "current-schema-id": 2   => ID points to latest schema
}
```



dremio

Ongoing & Future Work

- Iceberg View implementation: https://github.com/apache/iceberg/pull/4657
- Spark View Catalog
  - https://issues.apache.org/jira/browse/SPARK-31357
  - https://github.com/apache/spark/pull/35636
- Iceberg View changes for other Iceberg catalogs (Trino, Presto, Nessie, Glue, …)
- represent Views in an engine-agnostic way
  - https://substrait.io/ might help here
- support for Materialized Views in Iceberg?



Q & A